

In the Claims:

No claims are amended by this Reply. The claims are shown below for the convenience of the examiner.

1. (Previously Presented) A method for extending an existing object oriented programming language, comprising the steps of:

selecting a program source file including a workflow definition created using a workflow language, wherein the program source file includes a source code and classes therein and a workflow definition created using the workflow language that is specified in the form of annotations to the source code and the classes;

extending the source code with a plurality of workflow constructs, including an action construct representing an activity that allows a first software component to call an operation on a second software component; and

using a workflow program according to the workflow definition, including

processing, using a computer including a processing device operating thereon, the action construct to allow the first software component to call the operation on the second software component, and

passing, according to the workflow definition in the form of annotations to the source code, information selected from one or more files, documents and/or tasks between system resources, according to a set of procedural rules to generate activities at the computer as defined by the workflow definition.

2. (Previously Presented) A method according to claim 1, wherein:

the plurality of workflow definition constructs are provided as markup language commands that are then used as annotations to the source code and the classes.

3. (Previously Presented) A method according to claim 1, wherein:

the plurality of workflow definition constructs are provided as XML commands that are then used as annotations to the source code and the classes.

4-17. (Canceled)

18. (Previously Presented) A computer system capable of extending an existing programming language, comprising:

a computer including a processing device operating thereon;

a program source file stored on a computer readable storage medium, wherein the program source file includes a source code and classes therein;

a workflow definition created using the workflow language that is specified in the form of annotations to the source code and the classes, and wherein the workflow language extends the source code with a plurality of workflow constructs, including an action construct representing an activity that allows a first software component to call an operation on a second software component; and

object code executed by the processor, the object code configured to

use a workflow program according to the workflow definition, including processing, using a computer including a processing device operating thereon, the action construct to allow the first software component to call the operation on the second software component, and

pass information selected from one or more files, documents and/or tasks between system resources, according to a set of procedural rules to generate activities at the computer as defined by the workflow definition specified in the form of annotations to the source code and classes.

19. (Previously Presented) A computer system according to claim 18, wherein:

the plurality of workflow definition constructs are provided as markup language commands that are then used as annotations to the source code and the classes.

20. (Previously Presented) A computer system according to claim 18, wherein:

the plurality of workflow definition constructs are provided as XML commands that are then used as annotations to the source code and the classes.

21- 42. (Canceled)

43. (Previously Presented) A computer system according to claim 18, wherein the workflow definition is invoked by executing a software application.

44. (Previously Presented) A computer system according to 18, wherein the program source file is a Web Service file that includes the workflow definition constructs.

45. (Previously Presented) A computer system according to claim 44, wherein the workflow definition constructs of the Web Service file also references methods and variables for a software application running on the system and using the workflow.

46. (Previously Presented) A computer system according to claim 44, wherein the Web Service file includes the workflow definition constructs as a plurality of XML workflow annotations to the source code and classes defined in the Web Service file.

47. (Previously Presented) A computer system according to claim 46, wherein the XML workflow annotations to the source code and classes define a flow logic that can then reference the methods and variables defined in the Web Service file.

48. (Previously Presented) A computer system according to claim 44, wherein workflow definitions are provided as a separate Work Flow file that includes workflow definition commands, and that are invoked by the Web Service file using the workflow definition constructs, to use the workflow as defined by the Work Flow file.

49. (Previously Presented) A computer system according to claim 18, further comprising a light-weight virtual machine at the computer that processes the workflow and that is enabled to, at a particular point in the workflow process, save the workflow's execution space including program stack and variable state, and, at a later point in time, revive the workflow at the same point in the workflow process using the saved program stack and variable state.

50. (Previously Presented) A computer system according to claim 49, wherein the light-weight virtual machine is configured to
set the workflow program in a dormant condition; and
revive the dormant workflow program to its exact state before going dormant.

51. (Previously Presented) A computer system according to claim 18 including
a Java programming language as the program source file, wherein the Java programming language is extended by adding workflow constructs to said Java programming language, and
wherein said extending further comprises embedding the workflow constructs defined by XML in the Java programming language.

52. (Previously Presented) A method according to claim 1, wherein the workflow definition is invoked by executing a software application.

53. (Previously Presented) A method according to claim 1, wherein the program source file is a Web Service file that includes the workflow definition constructs.

54. (Previously Presented) A method according to claim 53, wherein the workflow definition constructs of the Web Service file also references methods and variables for a software application running on the system and using the workflow.

55. (Previously Presented) A method according to claim 53, wherein the Web Service file includes the workflow definition constructs as a plurality of XML workflow annotations to the source code and classes defined in the Web Service file.

56. (Previously Presented) A method according to claim 55, wherein the XML workflow annotations to the source code and classes define a flow logic that references the methods and variables defined in the Web Service file.

57. (Previously Presented) A method according to claim 53, wherein workflow definitions are provided as a separate Work Flow file that includes workflow definition commands, and that are invoked by the Web Service file using the workflow definition constructs, to use the workflow as defined by the Work Flow file.

58. (Previously Presented) A method according to claim 1, further comprising a light-weight virtual machine at the computer that processes the workflow and that is enabled to, at a particular point in the workflow process, save the workflow's execution space including program stack and variable state, and, at a later point in time, revive the workflow at the same point in the workflow process using the saved program stack and variable state.

59. (Previously Presented) A method according to claim 58, wherein the light-weight virtual machine is configured to set the workflow program in a dormant condition; and
revive the dormant workflow program to its exact state before going dormant.

60. (Previously Presented) A method according to claim 1 including
a Java programming language as the program source file, wherein the Java programming language is extended by adding workflow constructs to said Java programming language, and wherein said extending further comprises embedding the workflow constructs defined by XML in the Java programming language.

61. (Previously Presented) A computer readable storage medium including instructions stored thereon which when executed cause the computer to perform the steps of:

selecting a program source file including a workflow definition created using a workflow language, wherein the program source file includes a source code and classes therein and a workflow definition created using the workflow language that is specified in the form of annotations to the source code and the classes;

extending the source code with a plurality of workflow constructs, including an action construct representing an activity that allows a first software component to call an operation on a second software component; and

using a workflow program according to the workflow definition, including

processing, using a computer including a processing device operating thereon, the action construct to allow the first software component to call the operation on the second software component, and

passing, according to the workflow definition in the form of annotations to the source code, information selected from one or more files, documents and/or tasks between system resources, according to a set of procedural rules to generate activities at the computer as defined by the workflow definition.